UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/749,735	12/30/2003	Vladimir Savchenko	6570P066	9043

8791 7590 06/01/2009

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
PHAM, LUU T	

ART UNIT	PAPER NUMBER
2437	

MAIL DATE	DELIVERY MODE
06/01/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/749,735	SAVCHENKO ET AL.	
	Examiner	Art Unit	
	LUU PHAM	2437	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on *19 February 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *49-92* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *49-92* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *02/19/2009*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to the Amendment filed on 02/19/2009.

2. In the instant Amendment, Claims 1-48 were previously cancelled; Claims 49, 59, 71, and 83 have been amended; Claims 49, 59, 71, and 83 are independent claims. Claims 49-92 have been examined and are pending. **This Action is made FINAL.**

Response to Arguments

3. The objection to claims 52, 62, 74, and 86 are withdrawn as Applicant's arguments are persuasive.

4. The rejections of claims 71-92 under 35 U.S.C. § 101 are maintained as the claims being directed to non-statutory subject matter. Applicant's arguments with respect to the statutory subject matter of *"means for"* limitations recited in claim 71 have been fully considered by they are not persuasive. As claimed in the independent claim 71, *"A Web service provider"* comprising *"means for obtaining," "means for generating,"* and *"means for processing;"* and as discussed in the specification, ***"Web services are**, in general terms, **computer software**," (par. 0002)* and ***"Service provider 100**, may be, for example, a **Web application server that is implemented according to any of the Java 2 Enterprise Edition Specifications**;" (par. 0003);* (emphasis added). In addition to the above, as claimed in claim 59, steps of *"obtaining," "generating,"* and *"processing"* are performed by software instructions. It is clear that the aforementioned *"means for"* are implemented in software, which is non-statutory subject matter. Therefore, the claim is directed to non-statutory subject matter.

5. The rejections of claims 71-82 under 35 U.S.C. § 112, second paragraph, are maintained as the claims being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Applicant's arguments with respect to the structure for *"means for"* limitations have been fully considered but they are not persuasive. Applicant argues *"Figures 4-7 depicted and described in the Specification present the structures that support the means-plus-function limitations in the claims."* The Examiner respectfully disagrees. As depected in Figures 4-7 and described in the Specification *(pars. 0002-0007)*, the aforemnetioned *"means for"* are implemented in software, which is a non-statutory subject matter, and should not be considered as *"structure"* for said mean-plus-functions limitations. The claims are improper and found as indefinite because the claim recites *"means for"* language and there is no structure disclosed in the specification. *"If there is no structure in the specification corresponding to the means-plus-function limitation in the claims, the claims will be found invalid as indefinite."* Biomedino, LLC vs. Waters Technology Corp., 490 F.3d 946, 950 (Fed. Cir. 2007).

6. Applicants' arguments with respect to claims 49-92, regarding a graphical user interface having icons corresponding to a plurality of authentication protocols, have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. **Claims 71-92 are rejected under 35 U.S.C. 101** as being directed to non-statutory subject matter.

- **Regarding claim 71;** the claim is not directed to eligible subject matter. The claim recites *"means for obtaining a description of a Web service," "means for generating the Web service," "means for generating virtual interface,"* and *"means for processing message traffic;"* In light of the specification *(pars. 0002-0007)* and limitations claimed in claim 59, the aforementioned *"means for"* are implemented in software, which is non-statutory subject matter. Therefore, the claim is directed to non-statutory subject matter.

- **Regarding claims 75, 77, and 82;** claims 75, 77, and 82 recite *"means for processing message traffic," "means for generating a Web service,"* and *"means for obtaining a Web Service;"* These claims are also rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter for the same reasons.

- **Regarding claim 83;** claim 83 is rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter. Although the preamble of the claim recites "A *system,*" the body of the claim does not positively recited any elements of hardware. The body of the claim recites "*UDDI directory,*" "*Web service provider*" and *"Web service client;"* In light of the specification *(pars. 0002-0007),* "*UDDI directory,*" "*Web service provider*" and

"Web service client;" are implemented in software, which is non-statutory subject matter..

Therefore, the claim is not directed to eligible subject matter under 35 U.S.C. 101.

- **Regarding claims 84-92;** claims 84-92 are also rejected as non-statutory under 35 U.S.C. 101 for the same reasons.

Claim Rejections - 35 USC § 112

9. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

10. **Claims 71-82 are rejected under 35 U.S.C. 112, second paragraph,** as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

- **Regarding claims 71, 75, 77, and 82;** claims 71, 75, 77, and 82 have been found invalid as indefinite because the claims recite "*means for*" languages and there is no structure disclosed in the specification. *"If there is no structure in the specification corresponding to the means-plus-function limitation in the claims, the claims will be found invalid as indefinite." Biomedino, LLC vs. Waters Technology Corp.*, 490 F.3d 946, 950 (Fed. Cir. 2007)

- **Regarding claims 72-74, 76, and 78-81;** claims 72-74, 76, and 78-81 are dependent on claim 71, and therefore inherit the 35 U.S.C 112, second paragraph issues of the independent claim.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

13. **Claims 49-92 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Chappell et al., (hereinafter "Chappell"), "Java Web Services," published by O'Reilly in March 2002, in view of Sun Microsystems, (hereinafter "Sun"), "Building Web Service - Sun™ ONE Studio 5 Programming Series," published by Sun Microsystems, Inc., in June 2003.

- **Regarding claim 49**, Chappell discloses a method in a Web service provider communicatively interfaced with a plurality of Web service clients *(page 6, chapter 1)*, comprising:

obtaining a description of a Web service comprising protocol-independent business logic *(page 72, chapter 5: Web Services Description Language; section 5.1: Introduction to WSDL; "this extensibility allows WSDL to be used to describe endpoints and their messages regardless of the message format or network protocol used to exchange them.")*;

generating the Web service based on the description obtained, the generated Web service comprising the protocol-independent business logic in an executable format *(pages 18-19, Figs. 2.1 and 2.4; "with the service description in hand, the requester binds (i.e., creates a service request for) to a service."; pages 28-29, chapter 3: SOAP; pages 32-34, sections 3.5.2-3.5.3; generating SOAP object; page 72, section 5.1; SOAP is transport protocol-independent since it is able to encapsulate messages transmitted across a network using either HTTP, STMP, or FTP protocols; see also pages 138-156)*;

generating a first virtual interface to the Web service based on the description obtained *(pages 28-29, chapter 3: SOAP; pages 32-34, sections 3.5.2-3.5.3; generating SOAP object; pages 108-111; SOAP Envelope Interface is known as virtual interface; see also pages 173-174; Figs. 8.2-8.3)*, the first virtual interface comprising a mapping of the protocol-independent business logic of the Web service to a first transport protocol *(page 30, section 3.5: Anatomy of a SOAP Message; see also pages 72 and 86)*, wherein the first virtual interface to provide a first Web service client access to the protocol-independent

business logic of the Web service *(pages 86 and 88, section 5.2.6.2; see also pages 138-156 and 173-174; SOAP is transport protocol-independent since it is able to encapsulate messages transmitted across a network using either HTTP, STMP, or FTP protocols);*

processing message traffic exchanged between a first Web service client proxy associated with the first Web service client and the Web service via the first virtual interface in accordance with the first transport protocol *(page 9, Figs. 1.1-1.2; page 139; Fig. 7.1; pages 144-145, sections 7.1.5.1: 'Creating the message' to 7.1.5.3: 'Making the call'; pages 173-174; Figs. 8.2-8.3; "the SOAP handler creates SOAP envelope for the response and delivers the message to the client"; see also pages 138-156)*;

generating a second virtual interface to the Web service based on the description obtained *(pages 87-88, sections 5.2.6.1-5.2.6.2; XML tags <soap:binding>, <http:binding> and <mine:binding>)*, the second virtual interface comprising a mapping of the protocol- independent business logic of the Web service to a second transport protocol different than the first transport protocol *(pages 29-30, section 3.2- 3.5; "SOAP conversations to be carried out via a 'binding' to another lower-level protocol, and that binding would most likely be HTTP or SMTP"; pages 138-156 and 169-172; "SOAP is a wire protocol that can be layered upon other wire protocols such as HTTP, FTP, and SMTP; J2EE supports these Internet protocols through servlets." SMTP and FTP are known as second transport protocols; see also pages 87-88, sections 5.2.6.1-5.2.6.2; page 34, section: The SOAP Protocol Binding)*, wherein the second virtual interface to provide a second Web service client access to the protocol-independent business logic of the Web service without regenerating the Web service *(pages 29-30; pages 138-156 and 169-172;*

"SOAP is a wire protocol that can be layered upon other wire protocols such as HTTP, FTP, and SMTP; J2EE supports these Internet protocols through servlets."); and

processing message traffic exchanged between the second Web service client and the Web service via the second virtual interface in accordance with the second transport protocol, without regenerating the Web service *(pages 14-15, section 1.3: 'Web Services in a J2EE Environment'; Fig. 1.4; pages 130-152, section 7.1.7: Using JAXM for SOAP with Attachments; "the same receiver we used before generates much different results because we're now sending multipart message. Note the MINE boundaries that separate the message's part. The first part of the message is the SOAP envelope; the next two parts are the added attachment;" pages 169 and 173-174, Figs. 8.2 and 8.3; see also pages 138-156).*

Chappell does not explicitly disclose the first Web service client including a client protocol implementation further including a user selected authentication protocol, the client protocol implementation to be set by a graphical user interface having a plurality of icons representing a plurality of authentication protocols, the user selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol.

However, in an analogous art, Sun discloses a method for building Web Services, wherein the first Web service client including a client protocol implementation further including a user selected authentication protocol *(Sun: pages 194-213; Fig. A2: Authentication dialog box; user is able to select authentication option from authentication dialog box)*, the client protocol implementation to be set by a graphical user interface

having a plurality of icons representing a plurality of authentication protocols, the user selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol *(Sun: page 194-213; Figs. A2-A7, A10 and A14; Authentication dialog box and SSL/TLS Settings window)*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Sun with the method and system of Chappell wherein the first Web service client including a client protocol implementation further including a user selected authentication protocol, the client protocol implementation to be set by a graphical user interface having a plurality of icons representing a plurality of authentication protocols, the user selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol to provide users with a means for assigning authentication roles to a web service *(Sun: page 194)*.

- **Regarding claim 50,** Chappell and Sun disclose the method of claim 49.

Chappell further discloses the first transport protocol comprises an authentication protocol compatible with a message authentication type for the message traffic exchanged between the first Web service client and the Web service *(Chappell: pages 131-132, section 6.3.8: Security and Authentication; page 227-228; section 10.1: Incorporating Security Within XML; page 240, section 10.4.1: Digital Credentials Extensions to SOAP)*.

- **Regarding claim 51**, Chappell and Sun disclose the method of claim 50.

Chappell further discloses the message authentication type comprises an X.509 certificate authentication type based on an authentication protocol implementation of the first Web service client *(Chappell: pages 131-132, section 6.3.8: Security and Authentication; page 240, section 10.4.1: Digital Credentials Extensions to SOAP)*.

- **Regarding claim 52**, Chappell and Sun disclose the method of claim 49.

Chappell further discloses the first transport protocol is selected from the group comprising HyperText Transfer Protocol (HTTP), Simple Object Access Protocol (SOAP), SOAP over HTTP, SOAP over File Transfer Protocol (FTP), SOAP over Simple Mail Transfer Protocol (SMTP), and HTTP over Secure Socket Layer (HTTPS) *(Chappell: pages 22, section 2.1.2.3: 'Binding'; "web service description documents specify the network protocols (i.e., HTTP, MINE, SMTP, etc.) that a service supports;" pages 135-137, section 6.4: 'Using WSDL Definitions with UDDI'; see also pages 8, 88, and 169; "SOAP provides a standard packaging structure for transporting XML documents over variety of standard Internet technologies, including SMTP, HTTP, and FTP. It also defined encoding and binding standards for encoding non-XML RPC invocation in XML for transport.")*; and

wherein the second transport protocol is selected from the group comprising HTTP, SOAP, SOAP over HTTP, SOAP over FTP, SOAP over SMTP, and HTTPS, wherein the second transport protocol selected is different from the first transport protocol selected *(Chappell: pages 8, 88, and 169; "SOAP provides a standard packaging structure for transporting XML documents over variety of standard Internet technologies, including*

SMTP, HTTP, and FTP. It also defined encoding and binding standards for encoding non-XML RPC invocation in XML for transport.").

- **Regarding claim 53**, Chappell and Sun disclose the method of claim 49.

Chappell further discloses processing message traffic exchanged between the first Web service client and the Web service via a third virtual interface in accordance with a third transport protocol without regenerating the Web service *(Chappell: pages 8, 19, 22, 88, and 169; "SOAP provides a standard packaging structure for transporting XML document over variety of standard Internet technology, including SMTP, HTTP, and FTP." SMTP and FTP are known as third transport protocol)*, wherein the third virtual interface comprises a mapping of the protocol-independent business logic of the Web service to the third transport protocol *(Chappell: pages 29-30, section 3.2- 3.5; "SOAP conversations to be carried out via a 'binding' to another lower-level protocol, and that binding would most likely be HTTP or SMTP"; see also page 34, section: The SOAP Protocol Binding)*.

- **Regarding claim 54**, Chappell and Sun disclose the method of claim 49.

Chappell further discloses processing the message traffic exchanged between the first Web service client and the Web service via the first virtual interface comprises exchanging the message traffic with the Web service client through a Hyper Text Transfer Protocol (HTTP) proxy in an HTTP format *(Chappell: page 114; the UDDILookup class has static methods that create a dynamic Java proxy object that communicates using SOAP; pages 136-137, section 6.4.1; pages 158-159, sections 7.2.1-7.2.2)*.

- **Regarding claim 55**, Chappell and Sun disclose the method of claim 49.

Chappell further discloses generating a Web service client proxy responsive to a request, the Web service client proxy comprising the first virtual interface and the second virtual interface, wherein the Web service client proxy to execute at a Web service proxy server separate from the Web service provider *(Chappell: pages 158-164; "the client application has a local object, the "stub," that acts as a proxy for the remote object; the sub object has the same method as the remote object, but does not implement the business method.").*

- **Regarding claim 56**, Chappell and Sun disclose the method of claim 49.

Chappell further discloses the first transport protocol comprises an authentication mechanism *(Chappell: pages 131-132, section 6.3.8: Security and Authentication)* and a transport guarantee mechanism *(Chappell: page 155, section 7.1.9.1: ProviderConnectionFactory; GuaranteedMessaging parameter; page 198; HTTPR and ebXML are known as transport guarantee mechanisms).*

- **Regarding claim 57**, Chappell and Sun disclose the method of claim 56.

Chappell further discloses the first transport protocol further comprises a specified port binding *(Chappell: pages 72, 74, and 88; page 160, section 7.2.2.3: Generated Service Interface; xml tag <port>).*

- **Regarding claim 58**, Chappell and Sun disclose the method of claim 49.

Chappell further discloses obtaining the description of the Web service comprises obtaining a Web Service Definition Language (WSDL) document from a

Universal Description, Discovery, and Integration (UDDI) directory *(Chappell: pages 96-136, chapter 6: UDDI: Universal Description, Discovery, and Integration)*, the UDDI directory comprising a plurality of WSDL documents, each describing one of a plurality of Web services accessible via the Web service provider, wherein the WSDL document obtained describes the Web service comprising the protocol-independent business logic *(Chappell: pages 135-136; section 6.4: 'Using WSDL Definitions with UDDI')*.

- **Regarding claim 59**, Chappell discloses a Web service provider *(page 6, chapter 1)* comprising machine-readable medium having instructions stored thereon that, when executed by a processor, cause the processor to perform operations comprising:

obtaining a description of a Web service comprising protocol-independent business logic *(page 72, chapter 5: Web Services Description Language; section 5.1: Introduction to WSDL; "this extensibility allows WSDL to be used to describe endpoints and their messages regardless of the message format or network protocol used to exchange them.")*;

generating a first virtual interface to the Web service based on the description obtained *(pages 28-29, chapter 3: SOAP; pages 32-34, sections 3.5.2-3.5.3; generating SOAP object; pages 108-111; SOAP Envelope Interface is known as virtual interface; see also pages 173-174; Figs. 8.2-8.3)*, the first virtual interface comprising a mapping of the protocol-independent business logic of the Web service to a first transport protocol *(page 30, section 3.5: Anatomy of a SOAP Message; see also pages 72 and 86)*, wherein the first virtual interface to provide a first Web service client access to the protocol-independent business logic of the Web service *(pages 86 and 88, section 5.2.6.2; see also pages 138-*

156 and 173-174; SOAP is transport protocol-independent since it is able to encapsulate messages transmitted across a network using either HTTP, STMP, or FTP protocols);

processing message traffic exchanged between a first Web service client proxy associated with the first Web service client and the Web service via the first virtual interface in accordance with the first transport protocol *(page 9, Figs. 1.1-1.2; page 139; Fig. 7.1; pages 144-145, sections 7.1.5.1: 'Creating the message' to 7.1.5.3: 'Making the call'; pages 173-174; Figs. 8.2-8.3; "the SOAP handler creates SOAP envelope for the response and delivers the message to the client"; see also pages 138-156)*;

generating a second virtual interface to the Web service based on the description obtained obtained *(pages 87-88, sections 5.2.6.1-5.2.6.2; XML tags <soap:binding>, <http:binding> and <mine:binding>)*, the second virtual interface comprising a mapping of the protocol- independent business logic of the Web service to a second transport protocol different than the first transport protocol *(pages 29-30, section 3.2- 3.5; "SOAP conversations to be carried out via a 'binding' to another lower-level protocol, and that binding would most likely be HTTP or SMTP"; pages 138-156 and 169-172; "SOAP is a wire protocol that can be layered upon other wire protocols such as HTTP, FTP, and SMTP; J2EE supports these Internet protocols through servlets." SMTP and FTP are known as second transport protocols; see also pages 87-88, sections 5.2.6.1-5.2.6.2; page 34, section: The SOAP Protocol Binding)*, wherein the second virtual interface to provide a second Web service client access to the protocol-independent business logic of the Web service *(pages 29-30; pages 138-156 and 169-172; "SOAP is a wire protocol that can be*

layered upon other wire protocols such as HTTP, FTP, and SMTP; J2EE supports these Internet protocols through servlets."); and

processing message traffic exchanged between the second Web service client and the Web service via the second virtual interface in accordance with the second transport protocol *(pages 14-15, section 1.3: 'Web Services in a J2EE Environment'; Fig. 1.4; pages 130-152, section 7.1.7: Using JAXM for SOAP with Attachments; "the same receiver we used before generates much different results because we're now sending multipart message. Note the MINE boundaries that separate the message's part. The first part of the message is the SOAP envelope; the next two parts are the added attachment;" pages 169 and 173-174, Figs. 8.2 and 8.3; see also pages 138-156).*

Chappell does not explicitly disclose the first Web service client including a client protocol implementation further including a user selected authentication protocol, the client protocol implementation to be set by a graphical user interface having a plurality of icons representing a plurality of authentication protocols, the user selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol.

However, in an analogous art, Sun discloses a method for building Web Services, wherein the first Web service client including a client protocol implementation further including a user selected authentication protocol *(Sun: pages 194-213; Fig. A2: Authentication dialog box; user is able to select authentication option from authentication dialog box)*, the client protocol implementation to be set by a graphical user interface having a plurality of icons representing a plurality of authentication protocols, the user

selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol *(Sun: page 194-213; Figs. A2-A7, A10 and A14; Authentication dialog box and SSL/TLS Settings window)*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Sun with the method and system of Chappell wherein the first Web service client including a client protocol implementation further including a user selected authentication protocol, the client protocol implementation to be set by a graphical user interface having a plurality of icons representing a plurality of authentication protocols, the user selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol to provide users with a means for assigning authentication roles to a web service *(Sun: page 194)*.

- **Regarding claims 60-66**, claims 60-66 are similar in scope to claims 50-56 respectively, and are therefore rejected under similar rationale.

- **Regarding claim 67**, Chappell and Sun disclose the Web service provider of claim 59.

Chappell further discloses the first transport protocol comprises an encryption mechanism, the encryption mechanism to encrypt the message traffic exchanged between the first Web service client and the Web service via the first virtual interface *(Chappell:*

pages 228, 233-237, section 10.3: 'XML Encryption'; XML tag 'EncryptionMethod Algorithm').

- **Regarding claim 68**, Chappell and Sun disclose the Web service provider of claim 59.

Chappell further discloses the first transport protocol comprises a client session protocol to define a session feature for the message traffic exchanged between the first Web service client and the Web service via the first virtual interface *(Chappell: pages 172-177; "for servlets, conversations are associated with a sessionID variable that accompanies every HTTP request;" "the EJB can be stateless session or a stateful session.").*

- **Regarding claim 69**, Chappell and Sun disclose the Web service provider of claim 59.

Chappell further discloses the first transport protocol comprises a port binding, the port binding defining a communication port for the message traffic exchanged between the first Web service client and the Web service via the first virtual interface *(Chappell: pages 72, 74, and 88; page 160, section 7.2.2.3: Generated Service Interface; xml tag <port>).*

- **Regarding claim 70**, claim 70 is similar in scope to claim 58, and is therefore rejected under similar rationale.

- **Regarding claims 71-78**, claims 71-78 are similar in scope to claims 49-56 respectively, and are therefore rejected under similar rationale.

- **Regarding claims 79-81**, claims 79-81 are similar in scope to claims 67-69 respectively, and are therefore rejected under similar rationale.

- **Regarding claims 82**, claim 82 is similar in scope to claim 58, and is therefore rejected under similar rationale.

- **Regarding claims 83**, Chappell discloses a system comprising:

a Universal Description, Discovery, and Integration (UDDI) directory *(pages 96-136, chapter 6: UDDI: Universal Description, Discovery, and Integration)*, the UDDI directory comprising a plurality of WSDL documents, each describing one of a plurality of Web services *(pages 135-136; section 6.4: 'Using WSDL Definitions with UDDI')*;

a Web service provider executed by an application server communicatively interfaced with the UDDI directory and a plurality of Web service clients *(pages 19-25; section 2.1.3.1: 'Service Provider'; Fig. 2.1; pages 96-104, chapter 6: 'UDDI: Universal Description, Discovery, and Integration')*, wherein the Web service provider to:

obtain a WSDL document from the UDDI directory describing a Web service comprising protocol-independent business logic *(page 72, chapter 5: Web Services Description Language; section 5.1: Introduction to WSDL; "this extensibility allows WSDL to be used to describe endpoints and their messages regardless of the message format or network protocol used to exchange them.")*,

generate a first virtual interface to the Web service based on the WSDL document *(pages 28-29, chapter 3: SOAP; pages 32-34, sections 3.5.2-3.5.3; generating SOAP object; pages 108-111; SOAP Envelope Interface is known as virtual interface; see also pages 173-174; Figs. 8.2-8.3)*, the first virtual interface comprising a mapping of the protocol-independent business logic to a first transport protocol *(page 30, section 3.5: Anatomy of a SOAP Message; see also pages 72 and 86; pages 86 and 88, section 5.2.6.2; see also pages 138-156 and 173-174; SOAP is transport protocol-independent since it is able to encapsulate messages transmitted across a network using either HTTP, STMP, or FTP protocols)*, and

generate a second virtual interface to the Web service based on the WSDL document *(pages 87-88, sections 5.2.6.1-5.2.6.2; XML tags <soap:binding>, <http:binding> and <mine:binding>)*, the second virtual interface comprising a mapping of the protocol-independent business logic to a second transport protocol, different than the first transport protocol *(pages 29-30, section 3.2- 3.5; "SOAP conversations to be carried out via a 'binding' to another lower-level protocol, and that binding would most likely be HTTP or SMTP"; pages 138-156 and 169-172; "SOAP is a wire protocol that can be layered upon other wire protocols such as HTTP, FTP, and SMTP; J2EE supports these Internet protocols through servlets." SMTP and FTP are known as second transport protocols; see also pages 87-88, sections 5.2.6.1-5.2.6.2; page 34, section: The SOAP Protocol Binding)*;

a first Web service client communicably interfaced with the Web service provider via the first transport protocol, wherein the first Web service client associated with

a first Web service client proxy to send message traffic to the Web service via the first virtual interface at the Web service provider in accordance with the first transport protocol *(page 9, Figs. 1.1-1.2; page 139; Fig. 7.1; pages 144-145, sections 7.1.5.1: 'Creating the message' to 7.1.5.3: 'Making the call'; pages 173-174; Figs. 8.2-8.3; "the SOAP handler creates SOAP envelope for the response and delivers the message to the client"; see also pages 138-156);*

a second Web service client communicably interfaced with the Web service provider via the second transport protocol, wherein the second Web service client to send message traffic to the Web service via the second virtual interface at the Web service provider in accordance with the second transport protocol *(pages 14-15, section 1.3: 'Web Services in a J2EE Environment'; Fig. 1.4; pages 130-152, section 7.1.7: Using JAXM for SOAP with Attachments; "the same receiver we used before generates much different results because we're now sending multipart message. Note the MINE boundaries that separate the message's part. The first part of the message is the SOAP envelope; the next two parts are the added attachment;" pages 169 and 173-174, Figs. 8.2 and 8.3; see also pages 138-156).*

Chappell does not explicitly disclose the first Web service client including a client protocol implementation further including a user selected authentication protocol, the client protocol implementation to be set by a graphical user interface having a plurality of icons representing a plurality of authentication protocols, the user selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol.

However, in an analogous art, Sun discloses a method for building Web Services, wherein the first Web service client including a client protocol implementation further including a user selected authentication protocol *(Sun: pages 194-213; Fig. A2: Authentication dialog box; user is able to select authentication option from authentication dialog box)*, the client protocol implementation to be set by a graphical user interface having a plurality of icons representing a plurality of authentication protocols, the user selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol *(Sun: page 194-213; Figs. A2-A7, A10 and A14; Authentication dialog box and SSL/TLS Settings window)*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Sun with the method and system of Chappell wherein the first Web service client including a client protocol implementation further including a user selected authentication protocol, the client protocol implementation to be set by a graphical user interface having a plurality of icons representing a plurality of authentication protocols, the user selected authentication protocol to be established by an icon in the plurality of icons selected by the user that corresponds to the user selected authentication protocol to provide users with a means for assigning authentication roles to a web service *(Sun: page 194)*.

- **Regarding claims 84-87**, claims 84-87 are similar in scope to claims 50-53 respectively, and are therefore rejected under similar rationale.

- **Regarding claim 88**, Chappell and Sun disclose the system of claim 83.

Chappell further discloses a Web service proxy server separate from the Web service provider to receive a Web service client proxy from the Web service provider *(Chappell: page 95; "WASP can also dynamically access any J2EE resource, such as JMS Destination, JDBC driver, EJB, or J2EE CA adapter by creating a dynamic proxy.")*, the Web service client proxy comprising the first virtual interface and the second virtual interface, wherein the Web service client proxy to execute at the Web service proxy server *(Chappell: page 114; "the UDDILookup class has static methods that create a dynamic java proxy object that communicates using SOAP."; see also pages 136-137; ServiceRegistryProxy object)*.

- **Regarding claim 89**, claims 89 is similar in scope to claim 56, and is therefore rejected under similar rationale.

- **Regarding claims 90-92**, claims 90-92 are similar in scope to claims 67-69 respectively, and are therefore rejected under similar rationale.

Conclusion

14. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Luu Pham whose telephone number is 571-270-5002. The examiner can normally be reached on Monday through Friday, 7:30 AM - 5:00 PM (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Emmanuel L. Moise can be reached on 571-272-3865. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information

for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Luu Pham/
Examiner, Art Unit 2437

/Emmanuel L. Moise/
Supervisory Patent Examiner, Art Unit 2437